IN THE SPECIFICATION:

Please amend the paragraph beginning at page 8, line 4, as follows.

FIG. 1A is a schematic block diagram of a surface reconstruction system 100 in accordance with the present invention. The surface reconstruction system 100 is used for the acquisition of three-dimensional shape information. As shown in FIG. 1A, the surface reconstruction system 100 includes certain standard hardware components, such as a central processing unit 110, a memory subsystem ~~110~~ 114, an input/output (I/O) subsystem 118 and a graphics subsystem 112, all communicating through a system bus ~~112~~ 116. As discussed further below, data captured by sensors (not shown) in the acquisition system 120 are transferred through the input/output (I/O) subsystem 118 to memory for further processing.

Please amend the paragraph beginning at page 14, line 10, as follows.

Missing points create holes that cannot be filled by the pivoting ball. Any post processing hole-filling algorithm could be employed; in particular, the ball-pivoting algorithm can be applied multiple times, with increasing ball radii, as discussed below. ~~in a section entitled “Multiple Passes.”~~ When pivoting around a boundary edge, the ball can touch an unused point lying close to the surface. Again, surface normals are used to decide whether the point touched is valid or not, as shown in FIG. 2G. A triangle is rejected if the dot product of its normal with the surface normal is negative. Thus, as a result of missing data, the ball pivots around an edge until it reaches a sample that belongs to a different part of the surface. By checking that the triangle and data point normals are consistently oriented, the improper generation of a triangle is avoided.

Please amend the paragraph beginning at page 15, line 8, as follows.

Choosing a suitable value for the radius ρ of the pivoting ball is typically easy. Current structured-light or laser triangulation scanners produce very dense samplings, exceeding our requirement that intersample distance be less than half the size of features of interest. Knowledge of the sampling density characteristics of the scanner, and of the feature size to be captured, are enough to choose an appropriate radius. Alternatively, a small subset of the data could be analyzed to compute the point density.

An uneven sampling might arise when scanning a complex surface, with regions that project into small areas in the scanner direction. The best approach is to take additional scans with the scanner perpendicular to such regions, to acquire additional data. It is noted, however, that the ball-pivoting algorithm can be applied multiple times, with increasing ball radii, to handle uneven sampling densities, as described below. ~~in a section entitled "Multiple Passes."~~

Please amend the paragraph beginning at page 19, line 10, as follows.

FIG. 8B is a diagram illustrating how data points are stored in memory. Only points from scans whose bounding box 710 intersects the current slice are stored. As shown in FIG. 8B, the information 820 stored for each point includes a unique vertex identifier, the three coordinates of the point, its associated normal, a unique identifier of the scan to which the point belongs, and a boolean flag that indicates whether or not the point is part of the triangle mesh. The set of points is stored in a list 822. A three-dimensional voxel grid 824 (shown in two-dimensions in FIG. 8B for illustration) is used to index into the list 822 by location. Each cell, or voxel (volume element) in the grid 824 corresponds to a small cubic portion of space. Points contained in one voxel are stored contiguously in the list 822. Each voxel in the grid 824 stores a pointer to the first point in the list contained in the voxel. The set of points contained in the voxel extends from the one pointed by the voxel itself to the one before the point pointed to by the next voxel. In FIG. 8B, the representative voxel 830 contains points from ~~P1~~ PI to PK-1.

Please amend the paragraph beginning at page 20, line 13, as follows.

FIG. 9A is a diagram showing an example of the active-edge front. Multiple loops of edges are possible at any time during the construction of the triangle mesh. FIG. 9A shows two loops 950, 960. Loop 950 includes a sequence of edges and vertices. For example, two consecutive edges 920. 922 connect vertices 928, 930 and 932. The patches of surface inside loops 950 and 960 has been already triangulated. Each edge in the loop has one adjacent triangle. For example, edge 922 is adjacent to triangle 942, while ~~vertex~~ edge 920 is adjacent to triangle 940.